

### Nützliche Quellen

- \* **Buch:** "Barrierefreiheit verstehen und umsetzen", Jan Eric Hellbusch & Kerstin Probiesch, dpunkt.verlag, 2011, ISBN: 978-3-89864-864-6
- \* BITV Anforderungen
- \* BITV Prüfschritte
- \* MDN - Using ARIA
- \* MDN - ARIA Barrierefreiheit
- \* <http://www.access-for-all.ch/ch/barrierefreiheit/barrierefreies-webdesign/checklist-2.html>
- \* <http://www.barrierefreie-informationskultur.de>
- \* Zusätzliche Quelle für Lösungsideen <http://whatsock.com/tsg/>

### Screenreader

- \* NVDA Ein freier Screenreader für Windows Systeme.
- \* JAWS Ein kommerzieller Screenreader für Windows Systeme.
- \* TalkBack Ein Screenreader von google für Android Geräte (in der Regel vorinstalliert).
- \* ChromeVox Ein Screenreader von google für den Chrome Browser (in der Regel vorinstalliert).

### Werkzeuge zur Prüfung des Accessibility-Trees

Um zur Entwicklungszeit das Verhalten von Screenreadern halbwegs sicher erraten zu können, reicht eine Prüfung des DOMs nicht aus. Für die Ausgaben der Screenreader ist der Accessibility-Tree ausschlaggebend. Mit folgenden Werkzeugen lässt sich der Accessibility-Tree analysieren und prüfen:

- \* Active Accessibility Object Inspector (Windows)
- \* Aviewer (Firefox, IE und Chrome unter Windows)
- \* JAWS script namens "BX", das MSAA und UI Automation untersucht.
- \* Accessibility Inspector (OS X)
- \* AccProbe (Linux)
- \* Chrome: `chrome://accessibility` + Accessibility Developer Tools (Chrome Erweiterung)
- \* Chrome: Developer Tools (F12) / Elements / Untertab Accessibility
- \* Chrome Lighthouse testet automatisiert WCAG Suites{{/popup}}
- \* Axe API (Plugins für Lighthouse)
- \* Chrome Erweiterung Axe
- \* WAVE Browser Plugin
- \* WCAG Accessibility Audit Developer UI (Chrome Extension)
- \* Siteimprove Accessibility Checker (Chrome Extension)

### Quellen:

- \* <http://www.barrierefreies-webdesign.de/knowhow/msaa/tools.html>
- \* <https://medium.com/pulsar/which-accessibility-testing-tool-should-you-use-e5990e6ef0a>



By **Huluvu424242**  
(FunThomas424242)

Published 4th January, 2015.  
Last updated 28th July, 2019.  
Page 1 of 5.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

[cheatography.com/funthomas424242/](https://cheatography.com/funthomas424242/)  
[github.com/Huluvu424242](https://github.com/Huluvu424242)

### Struktur & Gliederung & Bezeichnung

#### Allgemeine Hinweise

\* Die Textstruktur, die Gliederung und auch Einrückungen müssen auch für nicht sehende Menschen klar erkennbar sein. Sie sollen die Orientierung, Navigation und den Lesefluß unterstützen.

\* Die Bezeichnungen für nicht sehende Nutzer sollten identisch zu denen für Sehende sein und nur um weitere Details angereichert werden. Hintergrund ist die Verständigung zwischen Sehenden und Nichtsehenden beim Betrachten genau der selben Seite am gleichen PC. Wenn der Sehende eine Schaltfläche "Als Gelesen markieren" sieht und der Screenreader über den Tooltip "Bearbeiten der Nachricht beenden" vorliest fällt es schwer auf die selbe Schaltfläche zu schließen.

#### Vorgehensweisen

\* Unterteilung der Seite in Regionen mit Hilfe von landmark roles. Diese lassen sich per Tastatur ansteuern.

\* Für Regionen die nicht über landmark roles abgedeckt werden können, lassen sich document structure roles verwenden. Diese sind nicht per Tastatur ansteuerbar sondern fügen der Seite nur nützliche Zusatzinformationen hinzu.

### Seitenregionen (aria landmark roles)

ARIA Role	HTML Element	Benutzung
role="application"	-	Sehr selten als landmark role nutzbar. Nur für Bereiche in denen eine Webanwendung mit eigenem Focusmanagement eingebettet ist. Wird häufig falsch verwendet, z.B. wenn lieber Widget Rollen eingesetzt werden sollten.
role="banner"	<header>	Region mit Infos über einen Seitenabschnitt - im Sinne einer Überschrift (je Seite mehrmals nutzbar).
role="contentinfo"	<footer>	Region mit Infos über einen Seitenabschnitt - im Sinne von Inhalt und Metadaten (je Seite mehrmals nutzbar).
role="complementary"	<aside>	Region für Abschnitte welche in Form von Inhaltsbausteinen zusätzliche, eigenständige Informationen bereitstellen.
role="form"	-	Region mit Formularelementen
role="main"	<main>	Hauptregion (je Seite nur eine möglich)
role="navigation"	<nav>	Region mit Navigationselementen
role="search"	-	Region mit Suchelementen

#### Allgemeine Hinweise

\* ARIA definiert 8 landmark roles von denen 5 sogar ein eigenes HTML Element besitzen. Die Verwendung des spezifischen HTML Elements mit der Rolle ist generell besser als ein DIV Element zusammen mit der Rolle.

\* Regionen sollen unterscheidbar sein, z.B. per aria-labelledby oder aria-label

**Quelle:** Strategie für Landmark Roles



By **Huluvu424242**  
(FunThomas424242)

Published 4th January, 2015.  
Last updated 28th July, 2019.  
Page 2 of 5.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

[cheatography.com/funthomas424242/](https://cheatography.com/funthomas424242/)  
[github.com/Huluvu424242](https://github.com/Huluvu424242)

### Dokumentstruktur (aria document structure roles)

ARIA Role	HTML Element	Verwendung
-----------	--------------	------------

article	<article>	Nutzbar für eigenständige Teilbereiche einer Seite, wie z.B. Artikel einer Zeitung oder Post in einem Weblog. Article sind <b>schachtelbar</b> . Screenreader können hier in einen Betrachtungsmodus wechseln. Daher evtl. nicht sinnvoll für Bereiche mit Interaktionselementen.
---------	-----------	---

columnheader

definition

directory

document

group

heading

img

list

listitem

math

note

presentation	-	Soll benutzt werden um Semantik zu entfernen, z.B. bei einer Layouttabelle soll dadurch der Eindruck verhindert werden es handle sich um eine Datentabelle mit Kopf und Body.
--------------	---	---

region	<section>
--------	-----------

row

rowgroup

rowheader

separator

toolbar

### Quellen:

\* [https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA\\_Techniques#Document\\_structure\\_roles](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques#Document_structure_roles)

\* [https://www.w3.org/TR/wai-aria/roles#document\\_structure\\_roles](https://www.w3.org/TR/wai-aria/roles#document_structure_roles)



By **Huluvu424242**  
(FunThomas424242)

Published 4th January, 2015.  
Last updated 28th July, 2019.  
Page 3 of 5.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

[cheatography.com/funthomas424242/](https://cheatography.com/funthomas424242/)

[github.com/Huluvu424242](https://github.com/Huluvu424242)

### Widget Aggregationen (aria composited roles)

ARIA Role	HTML Element	Verwendung
listbox	<select>	Realisierung von Auswahllisten

#### Quellen

- \* [https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA\\_Techniques#Composite\\_roles](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques#Composite_roles)
- \* [https://www.w3.org/TR/wai-aria/roles#composite\\_roles](https://www.w3.org/TR/wai-aria/roles#composite_roles)

### Widgets (aria widget roles)

ARIA Role	HTML Element	Verwendung
slider		
option		In Kombination mit listbox für Auswahllisten.

#### Quellen

- \* [https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA\\_Techniques#Widget\\_roles](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques#Widget_roles)
- \* [https://www.w3.org/TR/wai-aria/roles#widget\\_roles](https://www.w3.org/TR/wai-aria/roles#widget_roles)

### Spracheingabe für Input Felder

Die Möglichkeit der Spracheingabe ist nach der Nutzung des Vorlesens von Texten (text to speech) das Wichtigste Feature für Sehbehinderte Nutzer. Damit wird vieles vereinfacht.

#### Technische Umsetzung

```
<input x-webkit-speech />
```

Event nach Spracheingabe kann ausgewertet werden mit:

```
var sucheingabe = document.getElementsByTagName("input")[0];
sucheingabe.onwebkitspeechchange = function(e) {
location.href = "http://www.google.de/search?q="+ sucheingabe.value;
};
```

Quelle: <https://www.drweb.de/webkit-browser-formularfelder-mit-spracheingabe-versehen/>

(28.07.2019 funktioniert nur im Chrome)

### Tabellen (Datentabellen)

\* Tabellen benötigen eine Zusammenfassung über das was in der Tabelle dargestellt wird. Dazu wird das Element <caption> verwendet. Das frühere Attribut summary ist veraltet und darf nicht mehr verwendet werden.

\* Tabellenspalten müssen eine Überschrift besitzen, damit klar ist welcher Inhalt in der Spalte zu erwarten ist ohne vorher die ganze Tabelle sichten zu müssen. Realisiert wird dies durch Texte in den th Elementen. Falls sich die visuelle Repräsentation von der JAWS Repräsentation unterscheiden soll z.B. Mo und Montag im Datepicker kann ein span mit sr-only und ein span mit aria-hidden="true" im th genutzt werden.



By **Huluvu424242**  
(FunThomas424242)

Published 4th January, 2015.  
Last updated 28th July, 2019.  
Page 4 of 5.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

### Navigation und Interaktion

- \* Bei einer Navigation zurück soll der Fokus wieder genau dort stehen von wo die Navigation erfolgte z.B. auf Seite 3 in Zeile 7 auf dem 5. Link in einer mehrseitigen Liste (falls genau dieser Link vorher die Navigation auslöste).
- \* Bei Interaktionselementen welche je nach Status ihre Präsentation ändern z.B. ein Icon für erledigt und nicht erledigt muss nach Auslösen der Interaktion der Fokus wieder auf diesem Icon (Interaktionselement) stehen.

### Tooltips, Alternativtexte

- \* Nicht länger als 60 Zeichen sonst wird im strukturierten Modus von JAWS nur ein Teil des Textes auf dem Brailleviewer angezeigt.
- \* Textbeginn identisch mit der Oberfläche halten, damit Sehender und Nichtsehender sich über die Inhalte unterhalten können.
- \* Textanfänge in Listen möglichst mit unterschiedlichen Buchstaben oder Wortanfängen beginnen, damit der Nichtsehende über Tastenkürzel schnell die gewünschte Stelle erreichen kann (funktioniert nur bei Texten die nach einer Weile als bekannt angesehen werden können, z.B. Liste von Berufen etc.).

### Dialoge, Fragen und Meldungen

- \* Fragen sollen stets in einfachen kurzen Sätzen und positiv formuliert werden. Negierte Fragen sind schwerer verständlich und von Lernschwachen Menschen teilweise nicht korrekt erfassbar.
- \* Fehlermeldungen sollten prinzipiell aus 2 Sätzen bestehen. Im ersten Satz sollte die Fehlerursache kurz beschrieben werden. Der 2. Satz sollte einen Lösungsvorschlag beinhalten z.B. "Es wurde ein ungültiges Startdatum eingegeben. Bitte tragen Sie ein korrektes Datum als Startdatum ein."
- \* Fehlermeldungen wie "Für diese Aktion besitzen Sie keine Berechtigung" sollen vermieden werden - Ätschi bäh Effekt! Interaktionen die der Nutzer nicht durchführen darf sollen nicht angezeigt werden.

### Darstellung & Verhalten von Webseiten

- \* Zunächst wird die Struktur einer Webseite beschrieben z.B. durch HTML oder XML.
- \* Durch Frameworks (JSP, AngularJS, doxia, ...) wird die Struktur mit Dateninhalten und Verhaltensinformationen angereichert.
- \* Über Stylesheets (css, xsl) wird die Struktur mit Informationen zum Layout und Design angereichert.
- \* Über aria-role Attributen werden der Struktur semantische Informationen zugeordnet.

Das Gesamtergebnis im Browser ist eine Darstellung von Daten gemäß einem definierten Layout und entsprechend den Designvorgaben mit einem Verhalten welches sowohl durch die verwendeten Frameworks als auch über die per Rollen definierte Semantik bestimmt wird.



By **Huluvu424242**  
(FunThomas424242)

Published 4th January, 2015.  
Last updated 28th July, 2019.  
Page 5 of 5.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

[cheatography.com/funthomas424242/](https://cheatography.com/funthomas424242/)  
[github.com/Huluvu424242](https://github.com/Huluvu424242)